CLAIMS

What is claimed is:

1. A security data packet processing system comprising:

a transmitting (Tx) direct memory access (DMA) interface (314) receiving a streamed security data packet, selecting a channel for processing the streamed security data packet and transferring the streamed security data packet to an external memory;

an input DMA engine (306) retrieving portions of the streamed security data packet from the external memory after all portions of the streamed security data packet have been transferred to the external memory;

an input FIFO (308) receiving the portions of the streamed security data packet from the input DMA engine (306) in blocks of a predetermined byte size, portions being retained in a portion of the input FIFO allocated to the selected channel;

a context RAM (308) receiving a security association database (SAD) entry associated with the selected channel, the SAD entry being retrieved from the external memory by the input DMA engine; and

an input crypto DMA engine (310) providing the blocks of the security data packet to a processing engine for processing.

2. The system as claimed in claim 1 further comprising:

an output crypto FIFO (320) receiving processed blocks of the security packet from the processing engine;

an output DMA engine (322) transferring the processed blocks of the security packet to an external output memory (158); and

a receiving (Rx) direct memory access (DMA) interface (324) retrieving the processed blocks of the security packet from the external output memory (158) after all portions of the processed security data packet have been transferred to the external output memory (158), and transferring the processed blocks of the security data packet to a streaming interface for streaming.

3. The system as claimed in claim 2 wherein the receiving (Rx) DMA interface (324) includes a plurality of registers storing length information each of a plurality of processed security data packets, the receiving (Rx) DMA interface (324) performing the retrieving in response to the storing of the length information for an associated processed security data packet.

4. The system as claimed in claim 1 wherein the context RAM (308) includes a portion storing program state information associated with the selected channel.

5. The system as claimed in claim 1 wherein the transmitting (Tx) direct memory access (DMA) interface (314) selects a least busy channel based on an amount of buffer space available for a channel in the external memory (156).

6. The system as claimed in claim 1 wherein when the security packet is an outbound IPSec security packet and wherein an outer header (56) and IPSec header (55) are added to the outbound IPSec security packet when portions of the packet are buffered in input FIFO (308).

7. The system as claimed in claim 1 wherein when the security packet is an inbound IPSec security packet and wherein an outer header (66) and IPSec header (65) are removed from the outbound IPSec security packet prior to portions of the packet being buffered in input FIFO (308).

8. A method for processing a security data packet comprising:

receiving a streamed security data packet;

selecting a channel for processing the streamed security data packet;

transferring the streamed security data packet to an external memory;

retrieving portions of the streamed security data packet from the external memory after all portions of the streamed security data packet have been transferred to the external memory;

transferring the portions of the streamed security data packet in an input FIFO (308) from an input DMA engine (306) in blocks of a predetermined byte size, portions being retained in a portion of the input FIFO allocated to the selected channel;

receiving at a context RAM (308), a security association database (SAD) entry associated with the selected channel, the SAD entry being retrieved from the external memory by the input DMA engine; and

providing to an input crypto DMA engine (310) the blocks of the security data packet to a processing engine for processing.

9. The method as claimed in claim 8 further comprising:

receiving by an output crypto FIFO (320), processed blocks of the security packet from the processing engine;

transferring by an output DMA engine (322) the processed blocks of the security packet to an external output memory (158);

retrieving by a receiving (Rx) direct memory access (DMA) interface (324) the processed blocks of the security packet from the external output memory (158) after all portions of the processed security data packet have been transferred to the external output memory (158); and

transferring the processed blocks of the security data packet to a streaming interface for streaming.

10. The method as claimed in claim 9 further comprising storing length information for each of a plurality of processed security data packets in one of a plurality of registers of the receiving (Rx) DMA interface (324), and wherein the receiving (Rx) DMA interface (324) performs the retrieving in response to the storing of the length information for an associated processed security data packet.

11. The method as claimed in claim 8 further comprising storing program state information associated with the selected channel in a portion of the context RAM (308) for the selected channel.

12. The method as claimed in claim 8 further comprising selecting a least busy channel based on an amount of buffer space available for a channel in the external memory (156), the selecting being performed by the transmitting (Tx) DMA interface (314) .

13. The method as claimed in claim 8 wherein when the security packet is an outbound IPSec security packet, the method further comprises adding an outer header (56) and IPSec header (55) to the outbound IPSec security packet when portions of the packet are buffered in input FIFO (308).

14. The method as claimed in claim 8 wherein when the security packet is an inbound IPSec security packet, the method further comprises removing an outer header (66) and IPSec header (65) from the outbound IPSec security packet prior to portions of the packet being buffered in input FIFO (308).

15. A method of processing an IPSec security protocol packet, the IPSec security protocol packet comprising an IPSec header, the method comprising :

buffering an IPSec security protocol packet in an external memory;

reading portions of the buffered IPSec security protocol packet into a first local buffer, the portions having a predetermined number of bytes;

verifying header information of the IPSec security protocol packet;

reading a security association database (SAD) entry into the first local buffer;

processing the IPSec security protocol packet based on information in the SAD entry; and

storing the processed IPSec security protocol packet in an external memory.

16. The method as claimed in claim 15 further comprising parsing the IPSec header to retrieve a pointer to the SAD entry.

17. The method as claimed in claim 15 wherein prior to the processing step, the method includes prepending control information to the IPSec security protocol packet based on information the SAD entry, the control information for use in the processing step.

18. The method as claimed in claim 15 wherein the processing step includes performing a cryptographic operation on the IPSec security protocol packet, the cryptographic operation comprising either a decryption function or an authentication function when the IPSec security protocol packet is an inbound packet, and an encryption operation when the IPSec security protocol packet is an outbound packet.

19. The method as claimed in claim 15 further comprising selecting a least busy channel of a plurality of channels for processing the IPSec security protocol packet, and wherein the external memory has a portion associated with least busy channel.

20. The method as claimed in claim 15 wherein after the processing step, the method includes buffering the processed IPSec security protocol packet in a buffer allocated to the channel selected for the packet.

21. The method as claimed in claim 15 further comprising performing a security policy check on the processed IPSec security protocol packet, the security policy check comprising verifying that an IP source address is within a range of addresses identified by the SAD entry.

22. The method as claimed in claim 15 further comprising performing an anti-replay check on the processed IPSec security protocol packet, and updating a current byte count and anti-replay fields of the SAD entry.

23. An application specific integrated circuit for processing IPSec security protocol packets comprising:

a first streaming interface communicating with a network processor over a streaming interface and receiving a streamed packet;

an input buffer storing portions of the streamed packet along with control information for the packet;

a crypto core engine performing IPSec cryptographic operations on the packet in accordance with the control information;

an output buffer storing processed portions of the streamed packet; and

a second streaming interface receiving the processed portions of the streamed packet from the output buffer and providing the network processor a processed IPSec packet over the streaming interface.

24. The ASIC as claimed in claim 23 wherein the streaming interface selects a channel from a plurality of channels for processing the streamed packet, and wherein the input buffer and output buffer each have a portion thereof associated with each channel.

25. The ASIC as claimed in claim 24 further comprising a plurality of processing cores, each processing core associated with one of the channels and controlling the processing of an IPSec packet through the associated channel.

26. A method of processing data packets for implementing a security protocol, the method comprising:

receiving at a first streaming interface an IP data packet from a network processor, the IP data packet including a security association database (SAD) tag prepended thereto;

moving at least portions of the IP data packet in a first portion of a first buffer;

reading an SAD entry corresponding to the SAD tag into a second portion of the first buffer;

prepending control information to the IP data packet;

processing the IP data packet by performing a cryptographic operation on the IP data packet to generate a security protocol data packet; and

streaming the security protocol data packet from a second streaming interface to the network processor for transmission through the network.

27. The method as claimed in claim 26 wherein the security header and outer IP header are based on information from the corresponding SAD entry.

28. The method as claimed in claim 27 wherein the security protocol is an IPSec protocol, and wherein the security header is an IPSec header, and wherein the security protocol data packet is formatted in accordance with an IPSec security protocol.

29. The method as claimed in claim 26 wherein the cryptographic operation comprises either an encryption or authentication cryptographic operation, and wherein the method further comprising storing at least portions of the security protocol data packet in a second buffer.

30. The method as claimed in claim 26 further comprising the input streaming interface selecting a least busy channel from a plurality of channels for processing the IP data packet.

31. The method as claimed in claim 26 further comprising, prior to the reading, obtaining a semaphore for the SAD entry to prevent modification of data within the SAD entry by other channels.

32. The method as claimed in claim 31 further comprising, subsequent to the reading, updating a byte count and sequence number in the SAD entry.

33. The method as claimed in claim 26 wherein the storing comprises buffering the portions of the security protocol data packet, the portions comprising a predetermined number of bytes.

34. The method as claimed in claim 26 wherein the control information identifies an algorithm and key for the cryptographic operation to apply to the IP data packet.

35. The method as claimed in claim 26 further comprises checking a path maximum transmission unit (PMTU) value of the IP data packet including the security header and the outer IP header as prepended to the IP data packet to determine when the PMTU value exceeds a PMTU value for a tunnel through which the security protocol data packet is destined.

36. The method as claimed in claim 26 wherein the processing is performed by a crypto engine and wherein subsequent to the processing, the method further comprises prepending status information to the security protocol data packet, the status information being generated by the processing and identifying when the crypto engine detects an error.

37. The method as claimed in claim 26 wherein the streaming is performed when all portions of the security protocol data packet are stored in a second buffer.